ACCELERATION UNIT FOR HTTP HEADERS IDENTIFICATION IN FPGA

Ivan Bryndza

Bachelor Degree Programme (3), FEEC BUT

E-mail: xbrynd00@stud.feec.vutbr.cz

Supervised by: Marián Pristach

E-mail: xprist00@stud.feec.vutbr.cz

Abstract: This paper presents a hardware accelerated identification of HTTP protocol headers, since HTTP is the most used protocol on the Internet. We have designed a hardware architecture, which will be used for detection of HTTP header in each packet. Architecture will be able to achieve the throughput needed for monitoring of 100 Gb/s networks. Nondeterministic finite automata and massive parallelism is used for pattern match.

Keywords: HTTP, Nondeterministic Finite Automata (NFA), BRAM, Field Programmable Gate Array (FPGA)

1. ÚVOD

Kľúčovou a zároveň veľmi výpočtovo náročnou operáciou používanou v oblasti monitorovania a bezpečnosti počítačových sietí je hľadanie vzorov v dátach paketov. Vyhľadávanie vzorov je tatiež vhodné pre detekciu a spracovanie HTTP (Hypertext Transfer Protocol) protokolu, pričom je pre detekciu útokov potrebné hľadať rádovo tisíce vzorov na gigabitových rýchlostiach, čo súčasné softvérové riešenia neumožňujú. V posledných rokoch boli preto vytvorené hardvérové architektúry, založené na technológii FPGA, ktoré umožňujú hľadanie regulárnych výrazov v gigabitových linkách. Táto hardvérová architektúra je navrhnutá tak, aby dosahovala priepustnosť 100 Gb/s a zároveň zabrala menej zdrojov ako doposiaľ známe riešenia.

2. ARCHITEKTÚRA PRE DETEKCIU HTTP HLAVIČIEK

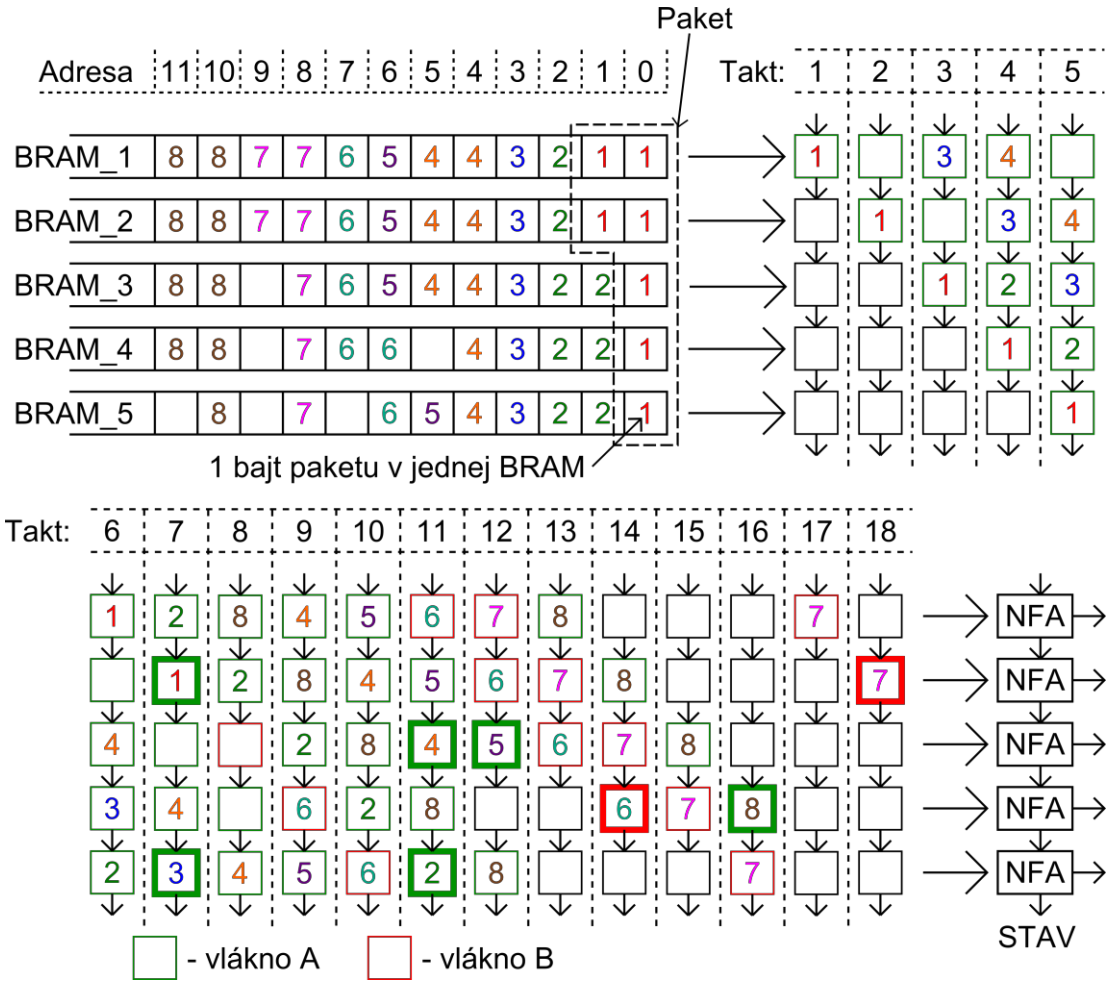
Pre detekciu nebezpečnej sieťovej komunikácie sú účinnejšie regulárne výrazy než reťazce [1]. Pomocou regulárnej množiny a regulárneho výrazu je možné reprezentovať ľubovoľný regulárny jazyk a naopak. Tento jazyk je možné popísať konečným stavovým automatom, pričom ku každému regulárnemu výrazu je možné vytvoriť odpovedajúci konečný automat, ktorý prijíma všetky reťazce definované regulárnym výrazom. Známy a dobre popísaný postup, umožňujúci hľadať množiny regulárnych výrazov, vychádza z mapovania nedeterministického konečného stavového automatu (*angl. nondeterministic finite automata*) (NFA) do technológie FPGA. Mapovanie NFA do FPGA rozšíril pán Clark o zdieľaný dekodér znakov [2].

NFA je popísaný tak, že spracováva jeden znak (jeden bajt) v jednom kroku. Prenosová rýchlosť tohto NFA je pri reálne dosiahnuteľnej frekvencii FPGA 200 MHz iba 1,6 Gb/s. Aby sa dosiahla žiadúca prenosová rýchlosť 100 Gb/s pri pracovnej frekvencii FPGA 200 MHz je potrebná zbernica so šírkou 512 bitov. Znamenalo by to zväčšenie vstupného zdieľaného dekodéru v NFA z 256 znakov na 2^{512} znakov, čo je exponenciálny nárast, nehovoriac o zložitosti NFA a počte jeho stavov, ktoré by bolo nutné popísať. Výsledný dizajn by zabral veľké množstvo zdrojov na čipe. Výhodnejší spôsob ako zabezpečiť prenosovú rýchlosť 100 Gb/s je použiť 64 jednoduchých NFA, ktoré spracovávajú 8 bitov (jeden bajt) v jednom takte. Spracovávalo by sa tak 8*64 bitov paralelne. Spôsobilo by to iba lineárny nárast zabratia zdrojov a zachovala by sa jednoduchosť NFA.

V navrhnutej architektúre preto pracujú NFA jednotky paralelne a každý automat spracováva v jednom takte iba 8 bitov. Na spracovaní každého paketu sa podieľajú všetky NFA jednotky, pričom každá spracuje jeden bajt a pošle informáciu NFA jednotke pod ňou. Tá spracuje ďalší bajt atď. Informácie o stave sa preposielajú v slučke až do spracovania celého paketu. Vyhľadávanie prebieha súčasne v niekoľkých paketoch, pričom každý paket je v inom stupni rozpracovania.

Základom architektúry pre detekciu HTTP protokolu v pakete je 64 blokových RAM (BRAM) pamätí, ktoré sú vstavané v FPGA. Na každú adresu sa uloží jeden bajt vstupného zbernicového slova. Jednému slovu teda pripadá rovnaká adresa vo všetkých BRAM. Následné čítanie z pamätí a posielanie do nedeterministických stavových automatov znázorňuje obrázok 1. Na obrázku 1 je pre zjednodušenie znázornených iba 5 BRAM pamätí. V týchto pamätiach je uložených 8 rôzne dlhých paketov.

V každom takte sa k prvej BRAM nastaví signál, ktorý určuje adresu vyčítania z BRAM a zároveň sa nastaví aj signál na určenie čísla BRAM, pri ktorej má čítanie paketu začať. Signály sa nastavujú stále iba k prvej BRAM pamäti. Ušetrí sa tak množstvo prepojení na čipe FPGA. V prvom takte sa nastaví vyčítanie z adresy nula hneď od prvej BRAM a začne sa spracovávať prvý paket. Spracovávanie prvého paketu pokračuje ďalej v ďalších taktoch, pričom sa vyčíta z adresy 0 postupne z každej BRAM. V druhom takte sa k prvej BRAM nastavia signály pre vyčítanie druhého paketu. Paket začína až v tretej BRAM na adrese jedna. Nastavia sa teda signály pre vyčítanie z adresy jedna od tretej BRAM. Všetky signály sa preposielajú v slučke pokým nie je celý paket spracovaný. Po spracovaní paketu sa slučka preruší a k prvej BRAM sa v danom takte nastavia nové signály. Na obrázku 1 je hrubo vyznačené spracovanie posledného bajtu paketu.



Obrázok 1: Spracovanie paketov

Príklad spracovania na obrázku 1 ukazuje, že 8 paketov sa spracovalo za 18 taktov. V prípade, že by sa pakety spracovávali iba jedným jednoduchým nedeterministickým stavovým automatom, spracovanie paketov na obrázku by zabralo 59 taktov. Je to výrazné urýchlenie z rýchlosti 1,6 Gb/s na rýchlosť 5,2 Gb/s, pri uvažovaní pracovnej frekvencie FPGA 200 MHz. Tento príklad má však len 5 BRAM pamätí. To znamená, že maximálna prenosová rýchlosť by v ideálnom prípade bola iba 8 Gb/s. Pri tejto rýchlosti by spracovanie všetkých paketov zabralo 12 taktov (5 bajtov v jednom takte). Veľké spomalenie spôsobuje „nábeh" spracovania a „dobeh" spracovania, spôsobený zreťazením výpočtov (pipeline). Vtedy sa takty nevyužívajú naplno. Príkladom je hneď prvý takt, kedy sa spracováva iba jeden bajt namiesto piatich bajtov naraz, ako je to v deviatom takte. Reálna komunikácia je však značne dlhšia a tak je vplyv týchto faktorov zanedbateľný. Ďalšie spomalenie spôsobujú rôzne dĺžky paketov a medzery medzi nimi. Tento faktor má len malý dopad na spomalenie priepustnosti.

Pre zmiernenie vplyvu rôzne dlhých paketov na spomalenie priepustnosti sa používajú dve vlákna (A a B). Príkladom urýchlenia je takt 6 na obrázku 1. V tomto takte sa z prvej BRAM číta prvý paket, ktorého spracovanie začalo už v prvom takte. Nové riadiace signály by sa v prípade jedného vlákna nemohli nastaviť. Šiesty paket by sa začal spracovávať až v jedenástom takte. V prípade využitia dvoch vlákien sa nové signály nastavia do vlákna B. Spracovanie šiesteho paketu sa spustí už v deviatom takte a jedenásty tak sa môže využiť pre spracovanie iného paketu.

3. ZÁVER

Výhodou tejto hardvérovej architektúry pre vyhľadávanie reťazcov typických pre HTTP, ale aj iných vzorov v pakete, je veľká úspora zdrojov a ich efektívne využitie pri dosiahnutí priepustnosti 100 Gb/s. Využitím paralelne pracujúcich nedeterministických stavových automatov (NFA) so zdieľaným dekodérom sa dosiahol iba lineárny nárast zabratia zdrojov a zachovala sa jednoduchosť stavového automatu. Nastavovaním signálov pre čítanie paketov vždy iba k prvej BRAM pamäti a ich automatickým preposielaním v každom takte k ďalším pamätiam sa ušetrilo množstvo prepojení na čipe FPGA, pretože nie je potrebné každú BRAM riadiť jednotlivo. Dizajn je navrhnutý pre kartu COMBO-100G [3], ktorá je vyvinutá organizáciou CESNET [4] v spolupráci s firmou INVEA-TECH [5]. Základom tejto karty je FPGA typu Virtex-7 H580T, ktoré disponuje 940 BRAM pamäťami, pričom sa na dosiahnutie priepustnosti 100 Gb/s využilo pre hlavnú funkciu iba 64 týchto pamätí. Zároveň je celý dizajn založený na generických parametroch pre budúce zvyšovanie priepustnosti, alebo prípadné znižovanie priepustnosti, pre jej nevyužívanie a potrebu úspory zdrojov na čipe. Ďalšou významnou skutočnosťou je, že celý dizajn je synchrónny, čiže pre svoju funkciu využíva iba jeden hodinový signál. Nie sú tak potrebné asynchrónne prechody medzi hodinovými doménami, ktoré by celý dizajn skomplikovali a spotrebovali ďalšie zdroje na čipe. Pre popis tejto architektúry sa využila výhoda HDL jazykov a to konkrétne jazyka VHDL, ktorou je možnosť popísať paralelne pracujúci systém.

REFERENCIE

[1] VÝZKUMNÁ SKUPINA ANT AT FIT. *Rychlé hledání regulárních výrazů*. Brno: 2010. Technická zpráva. FIT VUT.

[2] CHRISTOPHER R. CLARK AND DAVID E. SCHIMMEL. Efficient Reconfigurable Logic Circuits for Matching Complex. In: *Field Programmable Logic and Application, 13th International Conference*. Lisbon (Portugal): 2003, s. 956-59.

[3] Liberouter. *COMBO-100G* [online]. [cit. 2015-03-21]. Dostupné z: https://

[4] www.liberouter.org/combo-100g/

[5] *CESNET* [online]. [cit. 2015-03-21]. Dostupné z: http//:www.cesnet.cz/

[6] *INVEA-TECH* [online]. [cit. 2015-03-21]. Dostupné z: https://www.invea.com/en